

شرح أحجية وخوارزميته
(النسخة 0.5 بيتا 1)

غسان صبري أحمد السقاف
(سأتعلم من سوق الملح وأتفوق عليك!)



إلى عائلتي العزيزة
ومجتمع مسومس

وإلى كل شخص يؤمن بالعلم

المحتويات

المقدمة

بسم الله الرحمن الرحيم. والصلاة والسلام على أشرف خلق الله الصادق الأمين،
قال تعالى: "يرفع الله الذين آمنوا منكم والذين أوتوا العلم درجات"، (المجادلة، 11).

انطلاقًا من الآية الكريمة، ومن أهمية العلم ونشره لفائدة الجميع، قررت أن أرخص برنامج أحجية تحت رخصة غنو الرخصة العمومية، النسخة الثالثة (GNU/GPLv3)، ليتمكن الجميع من المشاركة والتعلم.

أحجية برنامج لتشفير الملفات النصية من نوع أسكي. يمكنك أن تفهم الأسكي (ASCII) على أنه النص الإنجليزي وبعض علامات الترقيم الأساسية.

جاءتني فكرة أحجية بينما كنت في حصة البرمجة، وكان الأستاذ يشرح عن خوارزميات الفرز في المصفوفات. إلا أنني في ذلك الوقت، ولضعف الأساسيات لدي في لغة الـ ++C، لم أتمكن من كتابة أحجية. كان هذا قبل حوالي خمسة أشهر من الآن؛ تحديدًا في جمادى الأول من هذا العام. الحمد لله، وبعد تعلمي وتعمقي في بعض الأساسيات في لغة الـ ++C رجعت إلي الفكرة التي لطالما شغلتني، وأراد الله سبحانه وتعالى فبدأت كتابة الملف المصدري.

إن شاء الله سأتطرق في هذا الكتيب الإلكتروني إلى شرح أحجية وخوارزميته، وكيفية اختيار كلمات المرور لتشفير أفضل، وبعض الملاحظات والأمور الأخرى المتعلقة بأحجية وخوارزميته. وفقني الله في ذلك.

في الأخير لا يسعني إلا أن أقول حجًا مبرورًا وذنبًا مغفورًا. سهل الله لنا حج البيت وزيارة الأراضي المقدسة.

غسان السقاف
03 - ذو الحجة – 1431هـ

أحجية: الفكرة الرئيسة للخوارزمية

تصادفك في بعض الأحيان ملفات مضغوطة محمية بكلمة مرور، ومتى أدخلت كلمة مرور خاطئة ظهرت لك رسالة تعلمك بذلك. "ماذا؟! وكيف يعلم أنني أدخلت كلمة مرور خاطئة؟!"، هكذا بدأ التساؤل يكبر شيئًا فشيئًا في ذهني. بدأت حينها أقرأ من شبكة المعلوماتية عن كيفية عمل خوارزمية البرامج التي تكشف كلمة المرور وتفك الملف المضغوط. فكانت الفكرة الرئيسة تتمثل في أن المبرمج يصمم برنامجًا يقوم بإنتاج كلمات مرور عشوائية ويتأكد منها بأن يمررها الملف ويرى النتيجة، فإن كانت خاطئةً سجلها في ملف لكي لا يعيد تكرارها، وينتج كلمات مرور جديدة، وهكذا.

"ماذا تقصد بالـ "النتيجة"؟"، قد تتساءل. في الحقيقة، لا أعلم. لكنني متأكد من أن البرنامج يتحقق كلمة المرور بسرعة، حتى لو كان الملف بالغيغا بايت. "إذًا، هو يتأكد من جزء بسيط من الملف فقط، ولا يمر على الملف كاملاً!"، هذا فكرت بها. بعبارة أبسط، لا يحتاج البرنامج لأن يمر بعملية معقدة لمعرفة إن كانت كلمة المرور خاطئة، وفقط يتأكد من جزء بسيط من الملف المضغوط.

إن برنامج أحجية مبني على فكرة [لا خطأ]، كما أحب تسميتها. "يجب أن يقبل أحجية أي كلمة مرور، وفي نفس الوقت يجب أن يكون قويًا كفاية لئلا يستطيع أحدٌ غير مرخص له أن يقرأ الرسالة"، هذه هي فكرة لا خطأ. "كيف يقبل أي كلمة مرور؟"، قد تتساءل. استنادًا إلى فكرة لا خطأ، فإن أحجية يقبل أي كلمة مرور، واستنادًا عليها يفك شفرة الملف. إذًا، سيفك التشفير، ويعطيك ناتج آخر مغاير عن الأصلي تمامًا. سأبسط فكرة لا خطأ في المثال التالي.

مثال:

إن كان مسعود قد اتصل بوالده وأخبره برقم معين (كلمة المرور)، وأخبره أن يضيف هذا الرقم إلى مجموع المال المشفر الذي أرسله له بالبريد الإلكتروني. إن كنت على علم بأن مجموع المال المشفر هو 100,000 ريال يمني، فكم من المال لدى مسعود؟

الحل:

والد مسعود ببساطة سيضيف الرقم (قد يكون الرقم موجبًا أو سالبًا) إلى المجموع المشفر، ويطلع بالنتيجة. أما أنت، فكيف ستعلم كم لدى مسعود؟

فإن افترضنا أن كلمة المرور كانت 100,000- فإن مسعود لم يجمع شيئًا؛ 100,000 – 100,000 = 0 ريال يمني.
أما أنت، فكيف ستعلم كم لدى مسعود؟

نرى بوضوح من المثال السابق كيف شُفرت كلمة المرور مع مجموع المال الفعلي في رقم واحد فقط! ومع ذلك، فإن باستطاعة الآخرين إنتاج أي كلمة مرور يحبونها، لكن النتيجة ستختلف. كما أن قوة لا خطأ تكمن في كيف سيعرفون أن الناتج خطأ؟ هذه فكرة لا خطأ!

أحجية: البرنامج

قبل أن أتطرق إلى الخوارزمية، سأشرح كيفية استخدام البرنامج.

في البداية، أحجية برنامج غير ذي واجهة رسومية (برنامج شاشة سوداء). لكن لا تهلع، صممته ليكون سهلاً لا غير! بتفكير قليل يمكنك أن تفهمه.

تشفير أم فك تشفير؟

عند بداية تشغيل أحجية، تظهر لك هذه الواجهة.

```
c:\Users\Ghasan\Desktop\C++\OhjyA\Release\OhjyA.exe
OhjyA is an ASCII-text file encryption program.
Copyright (C) 2010 Ghasan Al-Sakkaf.
OhjyA is licensed under GNU General Public License v3.
For detailed information, read 'COPYING.txt' & 'README.txt'.
---------

Do you want to Encrypt or Decrypt? '0' to Encrypt, and '1' to Decrypt.
_
```

الأربعة الأسطر الأولى توضح بعض المعلومات المتعلقة بأحجية.

في السطر الخامس، يخبرك أحجية ما إذا أردت التشفير (Encrypt) أو فك التشفير (Decrypt). أدخل [0] واضغط زر الإدخال [Enter] للتشفير، أو [1] لفك التشفير.

إن أدخلت أي شيء آخر، سيخبرك أحجية بوجود خطأ، ويطلب منك إعادة الإدخال كما اللقطة في الأسفل.

```
c:\Users\Ghasan\Desktop\C++\OhjyA\Release\OhjyA.exe
OhjyA is an ASCII-text file encryption program.
Copyright (C) 2010 Ghasan Al-Sakkaf.
OhjyA is licensed under GNU General Public License v3.
For detailed information, read 'COPYING.txt' & 'README.txt'.
---------

Do you want to Encrypt or Decrypt? '0' to Encrypt, and '1' to Decrypt.
3
Invalid input. Try again.

Do you want to Encrypt or Decrypt? '0' to Encrypt, and '1' to Decrypt.
```

التشفير

عند اختيارك للتشفير، ستظهر لك رسالة تفيدك بأن تدخل مسار الملف.

```
C:\Users\Ghasan\Desktop\C++\OhjyA\Release\OhjyA.exe
OhjyA is an ASCII-text file encryption program.
Copyright (C) 2010 Ghasan Al-Sakkaf.
OhjyA is licensed under GNU General Public License v3.
For detailed information, read 'COPYING.txt' & 'README.txt'.
----------

Do you want to Encrypt or Decrypt? '0' to Encrypt, and '1' to Decrypt.
0
Type path location of ASCII plain text file: _
```

تذكر دائمًا بأن مسار الملف لا يحتوي على نص عربي أو أي نص آخر غير الإنجليزية، فقط أسكي.
إن أدخلت المسار خطأً، أو أدخلت مسار ملف غير موجود فعلاً، ستظهر لك رسالة خطأ تعلمك بذلك، وتطلب منك إعادة إدخال المسار من جديد.
\\ تذكر دائمًا أن تدرج ملف نصي يحتوي فقط على نص أسكي، وأن يحتوي على الأقل على 10 محارف. (الفراغات والأسطر تسحب)

بعدها سيطلب منك أحجية أن تدخل كلمة المرور.

```
C:\Users\Ghasan\Desktop\C++\OhjyA\Release\OhjyA.exe
OhjyA is an ASCII-text file encryption program.
Copyright (C) 2010 Ghasan Al-Sakkaf.
OhjyA is licensed under GNU General Public License v3.
For detailed information, read 'COPYING.txt' & 'README.txt'.
----------

Do you want to Encrypt or Decrypt? '0' to Encrypt, and '1' to Decrypt.
0
Type path location of ASCII plain text file: C:\me.txt
Enter password:
```

طول كلمة المرور يجب أن يكون 6 محارف، ويجب أن يكون عبارة عن أرقام فقط من 0 إلى 9.
\\ يجب أن يكون طول كلمة المرور على الأقل 6 محارف، ولكنه غير محدود! (راجع صفحة [حيلٌ مفيدة])

إن كانت كلمة المرور موافقة للشروط، ولم يحدث أي خطأ، ستظهر لك هذه الرسالة.

```
C:\Users\Ghasan\Desktop\C++\OhjyA\Release\OhjyA.exe
OhjyA is an ASCII-text file encryption program.
Copyright (C) 2010 Ghasan Al-Sakkaf.
OhjyA is licensed under GNU General Public License v3.
For detailed information, read 'COPYING.txt' & 'README.txt'.
----------

Do you want to Encrypt or Decrypt? '0' to Encrypt, and '1' to Decrypt.
0
Type path location of ASCII plain text file: C:\me.txt
Enter password: 495847
Done!
Press any key to continue . . . _
```

لاحظ أن أحجية سيقوم بإنشاء ملف جديد في نفس المسار، وبنفس اسم الملف السابق إلا أنه سيضيف إليه امتداد أحجية [ohj.]، وإن كان الملف موجودًا بالفعل، فإن أحجية سيدمر ما بالملف من بيانات ويكتب عليها البيانات المشفرة.

بعدها اضغط أي زر، وسيخرج أحجية من تلقاء نفسه.

فك التشفير

عند اختيارك لفك التشفير، فإن هذه الرسالة ستظهر لك وتطلب منك إدخال مسار الملف المشفر.

```
C:\Users\Ghasan\Desktop\C++\OhjyA\Release\OhjyA.exe
OhjyA is an ASCII-text file encryption program.
Copyright (C) 2010 Ghasan Al-Sakkaf.
OhjyA is licensed under GNU General Public License v3.
For detailed information, read 'COPYING.txt' & 'README.txt'.
---------

Do you want to Encrypt or Decrypt? '0' to Encrypt, and '1' to Decrypt.
1
Type path location of OhjyA-encrypted file: _
```

عندها أدخل مسار الملف المشفر مراعيًا نفس الشروط المنطبقة عند التشفير.
\\ ليس ضروريًا أن يكون الملف المشفر بامتداد [ohj.]، لكن ذلك يجنبك بعض الأخطاء.
إن كان الملف قد عُدل وأضيفت له بعض الرموز غير الأرقام، فإن أحجية قد يرفضه.

إن قبل أحجية الملف، فإن هذه الرسالة ستظهر لك.

```
C:\Users\Ghasan\Desktop\C++\OhjyA\Release\OhjyA.exe
OhjyA is an ASCII-text file encryption program.
Copyright (C) 2010 Ghasan Al-Sakkaf.
OhjyA is licensed under GNU General Public License v3.
For detailed information, read 'COPYING.txt' & 'README.txt'.
---------

Do you want to Encrypt or Decrypt? '0' to Encrypt, and '1' to Decrypt.
1
Type path location of OhjyA-encrypted file: C:\me.txt.ohj
Enter password: 495847
Done!
Press any key to continue . . . _
```

أدخل عندها كلمة المرور مراعيًا نفس الشروط المطبقة عند التشفير.

إن كانت كلمة المرور التي أدخلتها موافقة للشروط وإن حتى خاطئة، فإن أحجية سيقبلها ويمررها على بعض العمليات، وستختلف النتيجة. (راجع صفحة [أحجية: الخوارزمية] لتفهم الخوارزمية).

سيقوم بعدها أحجية بإنشاء ملف في نفس المسار الذي أدخلته، وله نفس اسم الملف محذوفًا منه الامتداد [ohj.]. إن كان الملف موجودًا بالفعل، فإن البرنامج سيدمر ما بداخله ويعيد الكتابة عليها.

بعدها اضغط أي زر، وسيخرج أحجية من تلقاء نفسه.

أحجية: الخوارزمية

إن برنامج أحجية يحمل بذاته خوارزمية فريدة، حتى أنني في بعض الأحيان أطلق أحجية على الخوارزمية، وليس على البرنامج. لأن الخوارزمية بحد ذاتها أحجية!

إعادة التعريف

في البداية يقوم أحجية بإعادة تعريف المحارف كالتالي:

- **Z-A يتم تعريفها رقميًا من 11 – 36؛ فـ A = 11 تصاعديًا حتى Z = 36.**
- **z-a يتم تعريفها رقميًا من 37 – 62؛ فـ a = 37 تصاعديًا حتى z = 62.**
- **9 – 0 يتم تعريفها رقميًا من 63 – 72؛ فـ 0 = 63 تصاعديًا حتى 9 = 72.**
- **المحارف الأخرى يتم تعريفها كالتالي:**
  - **[\n] (زر الإدخال\السطر الجديد) = 73؛**
  - **[ ] (الفراغ) = 74؛**
  - **[؟] (الاستفهام) = 75؛**
  - **[!] (التعجب) = 76؛**
  - **[&] (الربط) = 77؛**
  - **[:] (نقطتان فوق بعض) = 78؛**
  - **[.] (نقطة) = 79؛**
  - **[,] (الفاصلة) = 80؛**
  - **[;] (الفاصلة المنقوطة) = 81؛**
  - **[)] (القوس الأيسر) = 82؛**
  - **[(] (القوس الأيمن) = 83؛**
  - **[/] (خط مائل إلى اليمين) = 84؛**
  - **[+] (علامة الجمع) = 85؛**
  - **[-] (علامة الطرح) = 86؛**
  - **[*] (علامة الضرب) = 87؛**
  - **['] (علامة تنصيص منفردة) = 88؛**
  - **[\] (خط مائل إلى اليسار) = 89؛**
  - **[عدا ذلك] (أي محرف آخر) = 90**

\\ لاحظ أن أي محرف خارج التعريف يتم إعطاءه نفس القيمة. ستعرف لاحقًا لماذا أوقفت التعريف عند 90.

الجمع

قبل أن أتقدم، رأيت أنه من الجيد أضمن مثالاً أشرح عليه ليسهل الفهم ويتضح.
لنفترض أن الملف احتوى على هذا النص: "ABCDEFGHIJ"، فإن إعادة التعريف ستنتج سلسلة التعريف هذه "11121314151617181920". ولنفترض أيضًا أن كلمة المرور التي أدخلها المستخدم "123456" .

يقوم أحجية بعد إعادة التعريف باللف على العناصر في سلسلة التعريف، ومقابلها في سلسلة كلمة المرور ويجمعهما مع بعضهما. بالاستناد إلى المثال السابق، فإن أحجية سيلف 10 مرات على سلسلة التعريف وفي نفس الوقت يلف على كلمة المرور كالتالي:

1. يأخذ 11 + 1 = 12
2. يأخذ 12 + 2 = 14
3. يأخذ 13 + 3 = 16
4. يأخذ 14 + 4 = 18
5. يأخذ 15 + 5 = 20
6. يأخذ 16 + 6 = 22
7. يأخذ 17 + 1 = 18 (لاحظ كيف أعاد أحجية اللف على كلمة المرور عند وصوله النهاية)
8. يأخذ 18 + 2 = 20
9. يأخذ 19 + 3 = 22
10. يأخذ 20 + 4 = 24

بعد ذلك تصبح سلسلة التعريف "12141618202218202224".

العاصفة

تقنية [العاصفة] هي القوة الضاربة في أحجية، والتي تغير مواقع العناصر بشكل مذهل استنادًا على قيمة العنصر في سلسلة كلمة المرور.

كما رأينا حتى الآن، يتم التعامل مع العناصر في سلسلة التعريف على شكل رقم من قوة العشرات. العاصفة تتعامل مع سلسلة التعريف كأرقام منفردة؛ مثلاً 20 تقسمه إلى قسمين 0 و 2 وتعاملهما بانفراد تام. وتقوم بعدها بنقلهما إلى أماكن أخرى في سلسلة التعريف معتمدة على قيمة كلمة المرور التي تلف عليها.

تذكر، حتى الآن سلسلة التعريف "121416182022182022224"، إذًا لمنضِ! هاك العملية:

1. ينقل 1 خطوة واحدة لأن 1 خطوة واحدة: 211416182022182022224
2. ينقل 1 خطوتين لأن 2 خطوتين: 241116182022182022224
3. ينقل 1 ثلاث خطوات لأن 3 ثلاث خطوات: 246111182022182022224
4. ينقل 1 أربع خطوات لأن 4 أربع خطوات: 246811112022182022224
5. ينقل 1 خمس خطوات لأن 5 خمس خطوات: 246801112122182022224
6. ينقل 1 ست خطوات لأن 6 ست خطوات: 246802112121182022224
7. ينقل 1 خطوة واحدة لأن 1 خطوة واحدة: 246802112121182022224
8. ينقل 1 خطوتين لأن 2 خطوتين: 246802112121182022224
9. ينقل 2 ثلاث خطوات لأن 3 ثلاث خطوات: 246802111122182022224
10. ينقل 1 أربع خطوات لأن 4 أربع خطوات: 246802111822112022224
11. ينقل 2 خمس خطوات لأن 5 خمس خطوات: 246802111802112222224
12. ينقل 2 ست خطوات لأن 6 ست خطوات: 246802111802112222224
13. ينقل 1 خطوة واحدة لأن 1 خطوة واحدة: 246802111802112222224
14. ينقل 1 خطوتين لأن 2 خطوتين: 246802111802122122224
15. ينقل 2 ثلاث خطوات لأن 3 ثلاث خطوات: 246802111802122122224
16. ينقل 1 أربع خطوات لأن 4 أربع خطوات: 246802111802122242221
17. لا ينقل 2 خمس خطوات لأن 5 خارج السلسلة: 246802111802122242221
18. لا ينقل 2 ست خطوات لأن 6 خارج السلسلة: 246802111802122242221
19. ينقل 2 خطوة واحدة لأن 1 خطوة واحدة: 246802111802122242212

لاحظ كيف أن اللف على سلسلة كلمة المرور يستمر وإن حتى لم تنفذ قيمة أحد العناصر في العملية لوقوعها خارج السلسلة.

أصبحت الآن سلسلة التعريف "246802111802122242212".

قناع الأرقام

قناع الأرقام تقنية تقوم باستبدال الأرقام في سلسلة التعريف بأرقام أخرى أعيد ترتيبها باستخدام تقنية العاصفة في مصفوفة من 10 عناصر تتضمن الأرقام من 0 إلى 9.

في اللغات البرمجية تبدأ المصفوفات من الموقع 0، لذا قمت بعمل مصفوفة رقمية من 10 أرقام من 0 إلى 9، بحيث الموقع 0 يتضمن 0 والموقع 1 يتضمن 1، كما هو موضح في الجدول التالي:

9	8	7	6	5	4	3	2	1	0
9	8	7	6	5	4	3	2	1	0

يقوم أحجية بتطبيق تقنية العاصفة على مصفوفة الأرقام كالتالي:

1. ينقل 0 خطوة واحدة لأن 1 خطوة واحدة: 1023456789
2. ينقل 0 خطوتين لأن 2 خطوتين: 1320456789
3. ينقل 2 ثلاث خطوات لأن 3 ثلاث خطوات: 1350426789
4. ينقل 0 أربع خطوات لأن 4 أربع خطوات: 1357426089
5. ينقل 4 خمس خطوات لأن 5 خمس خطوات: 1357926084
6. لا ينقل 2 ست خطوات لأن 6 خارج السلسلة: 1357926084
7. ينقل 6 خطوة واحدة لأن 1 خطوة واحدة: 1357920684
8. ينقل 6 خطوتين لأن 2 خطوتين: 1357920486
9. لا ينقل 8 ثلاث خطوات لأن 3 خارج السلسلة: 1357920486

بالعد من اليسار إلى اليمين، تغير الآن الجدول كالتالي:

6	8	4	0	2	9	7	5	3	1
9	8	7	6	5	4	3	2	1	0

من الجدول، كل عنصر من سلسلة التعريف يتحول إلى القناع الذي وسم به في مصفوقة الأرقام. من ذلك، تتحول سلسلة التعريف "24680211180212242212" إلى "59081533381535595535".

ينتهي بذلك أحجية من التشفير، ويكتب سلسلة التعريف بداخل الملف المشفر.

فك التشفير هو عملية عكسية للتشفير، إلا أن الاختلاف يكمن في أن المحارف غير المعرفة ستعامل كأنها حرف واحد [@]. هذه العلامة ستستبدل بأي محرف غير معرف.

أوقفت التعريف عند 90 لأن أكبر رقم يستطيع المتسخدم إدخاله هو 9، وإن حدث وجمعا فستكون النتيجة 99. ولكن إن زدت التعريف إلى 91 مثلاً، وجمع مع 9 فإن الناتج سيكون 100، وسيخرج عن كونه رقمًا من قوة العشرات إلى قوة المئات، وسيؤدي ذلك إلى ظهور كثير من الأخطاء في تقنية العاصفة؛ حيث أنها تتعامل، على الأقل حتى هذه اللحظة، مع الأرقام من قوة العشرات فقط.

أرجو أن يوفقني الله، وأن ييسر لي أمري. كما أثق بأن المبدعين لن يتوقفوا عن التطوير، فقد رخصت البرنامج تحت رخصة غنو الرخصة العمومية ليتمكن الجميع من المشاركة والتعلم!

أحجية: المميزات والعيوب

المميزات:

- السرعة: يتميز أحجية بسرعة في التشفير وفكه، وتظهر قوته جليةً في الملفات الكبيرة.
- لا خطأ: هذه التقنية إضافة قوية لأحجية، بحيث إن أراد أحدٌ فك التشفير سيحتاج لأن يلف على الملف كاملاً، ثم يصمم برنامج ليتعرف إن كان الناتج نصًا مفهومًا. (طريقة التعرف على النص غير مضمومنة. راجع صفحة [حيلٌ مفيدة] لتتعرف على كيفية التغلب على هذه النقطة)
- مكتوب بالـ C++ الأساسية (Native C++) والتي تمكن إعادة بنائه على أنظمة أخرى غير الويندوز!
- مفتوح المصدر، مرخص تحت رخصة غنو الرخصة العمومية، النسخة الثالثة (GNU/GPLv3). (راجع "COPYING.txt" المرفق مع حزمة أحجية لمعلومات أكثر تفصيلاً)
- خوارزمية فريدة من نوعها وقوية في نفس الوقت، وقابلة للتوسع أيضًا! (لأكون صادقًا، في رأسي الآن فكرة جهنمية، قد تُنفذ عند إطلاق النسخة المستقرة رأس السنة الميلادية القادمة)
- يمكن أن يضمن أحجية في مشاريع أخرى مثل برامج المحادثة و مواقع البروكسي، على سبيل المثال لا الحصر.
- إن رأيت فيه ضعفًا، فطوره بنفسك ولا تنتظر أحدًا ليفعل ذلك. فببساطة، لديك الكثير!

العيوب:

- تشفير الملفات الصغيرة أضعف من تشفير الملفات الكبيرة؛ كلما كبر حجم الملف المراد تشفيره ازدادت قوة التشفير.
- إذا أدخل المستخدم أرقامًا متتابعة مثل: 123456 و 456789 خاصة في الملفات الصغيرة، فإن فرص فك تشفير الملف تزداد. (لا بأس بـ 124567 أو 456879)
- إذا أدخل المستخدم كلمة مرور ملفوفة على نفسها مثل: 6848968489 و 73637363، فإن ذلك يعني كما لو أن المستخدم أدخل 68489 و 7363 بالترتيب، حيث أن البرنامج يلف من تلقاء نفسه على سلسلة كلمة المرور.

حيلٌ مفيدة

بعض عيوب أحجية يمكن تجاوزها ببعض الحيل كالتالي:

- إن كان الملف الذي تريد تشفيره كبيرًا، يفضل أن تجعل كلمة المرور طويلة. لكن في نفس الوقت لا تكتب كلمة مرور ضعيفة كما هو موضح في عيوب أحجية.
- لمنع برامج التعرف على النص من التعرف على ناتج الملف بسهولة، قم بكتابة رسائلك بلغة الرسائل القصيرة؛ فبدلاً عن كتابة: Wait for me at the foyer، اكتب: w8 4 me @ foyr، وأضمن لك تشفيرًا مرعبًا!
- يمكنك أيضًا أن تشفر الملف المشفر من جديد، ولكن يفضل أن تغير كلمة المرور ولو رقمًا واحدًا. بحيث ستحتاج لأن تفك شفرته مرتين لتحصل على النص الأصلي.

أحجية: ما بعد 0.5 بيتا 1

لن أتوقف - إن شاء الله - عن تطوير أحجية، فهذه بالنسبة لي مجرد البداية. فأنا عازمٌ على تطويره بمشاركة مطورين آخرين من على وجه الأرض.

قد أكون كالأطفال إن قلت أنني أسعى جاهدًا لأن يكون أحجية أقوى برنامج تشفير نصي، بل وتتطور فكرته ويتم تطبيقها في تشفير الملفات المختلفة (غير النصية)، إلا أنني من كل قلبي أتمنى ذلك، بل وسأطبقه واقعًا إن شاء الله!

في الوقت الحالي وحتى رأس السنة الميلادية القادمة، سأكثف تركيزي على تطوير أحجية، وتجاوز عيوب كلمة المرور، وإن شاء الله سأنجح، لأنه وببساطة: "إن الله لا يضيع أجر من أحسن عملاً".

كلمات تقدير

إن أحجية ليس فقط نتاج جهد ذاتي، بل ولأنني، بطريقة أو بأخرى، تأثرت بأناس أثاروا في روح الجد وعدم الاستسلام.

أود أن أتقدم بخالص التقدير إلى:

- عائلتي العزيزة فردًا فردًا، متمنيًا لهم التوفيق والسداد دنيًا وأخرى.
- محمد الأصبحي، رفيق الدرب.
- محمد المرهبي، صديقي العزيز في كلية الحاسوب في جامعة صنعاء.
- وكل فرد من مجتمع مسومس المبدع!

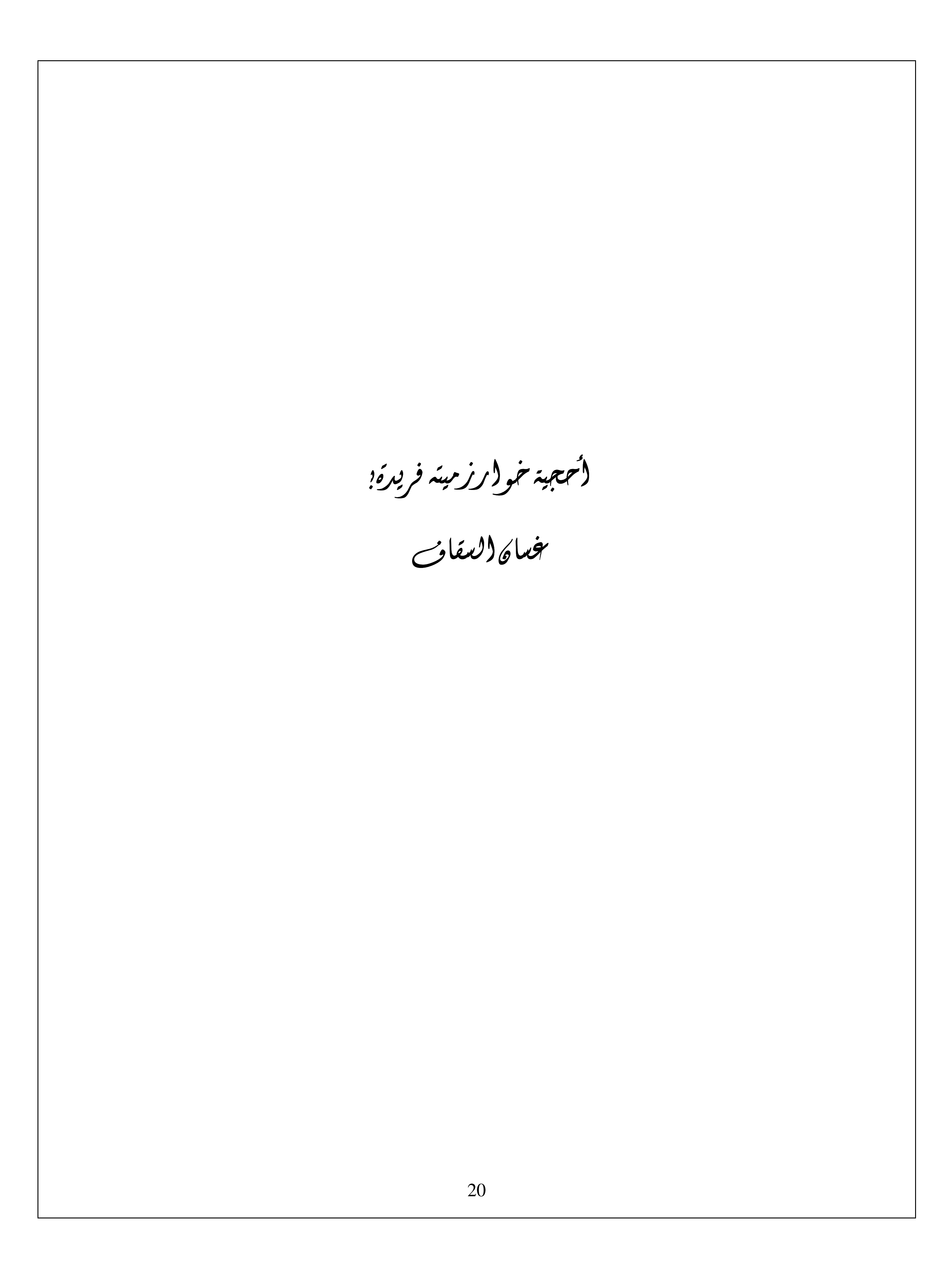
الخاتمة

ليس لدي الكثير لأقوله حقيقة، إلا أنني أريد أن أشجع على التعلم فالتعليم، والفشل فالتطوير. يجب علينا أن نزيد الحياة رفعة، ذلك لأننا وببساطة مسلمون، وهذا واجبنا.

وأحب أن أذكر من يرى أن العلم توقف عنده بقول الإمام الشافعي:

لن يبــلغ العلــم جمــيعًا أحــدٌ - لا ولو حاول ألف سنـــــــةٍ
إنـــما العلــم عمـــــيقٌ بحــره - فخــذوا من كل شــيءٍ أحسنه

"يرفع الله الذين آمنوا منكم والذين أوتوا العلم درجات"



أحجية خوارزمية فريدة!

غسان السقاف